

⑨ 日本国特許庁 (JP)

⑩ 特許出願公開

⑪ 公開特許公報 (A) 昭64-36336

⑤ Int. Cl.

G 06 F 9/46
9/38

識別記号

3 1 3
3 3 0

府内整理番号

C-7056-5B
K-7361-5B

④ 公開 昭和64年(1989)2月7日

審査請求 有 請求項の数 1 (全14頁)

⑤ 発明の名称 計算機システム

⑥ 特願 昭63-122246

⑦ 出願 昭63(1988)5月20日

優先権主張 ⑧ 1987年7月20日⑨ 米国(US)⑩ 075483

⑪ 発明者 ジョン・ステイフェン・リップタイ アメリカ合衆国ニューヨーク州リーヴィングス、トロイ・ドライブ1番地

⑫ 出願人 インターナショナル・ビジネス・マシーンズ・コーポレーション アメリカ合衆国10504、ニューヨーク州 アーモンク(番地なし)

⑬ 代理人 弁理士 頤宮 孝一 外1名

明細書

1. 発明の名称 計算機システム

2. 特許請求の範囲

システムのアーキテクチャにより要求されるアドレス可能なレジスタの数を n としたとき、最も大きな n 個のハードウェア・レジスタを有するレジスタ・アレイと、

上記レジスタ・アレイに情報を送り且つ上記レジスタ・アレイから情報を受け取るように接続された、命令を実行する命令実行装置と、

命令及びデータを保持し且つ上記命令実行装置に命令及びデータを供給する記憶装置と、

上記記憶装置及び上記命令実行装置に接続され、上記記憶装置から上記命令実行装置への命令の移動を制御する命令制御装置と、

上記レジスタ・アレイに接続され、命令実行時にアドレス可能なレジスタに上記ハードウェア・レジスタを割り当てる、且つ順序から外れた命令の実行時には選択されたアドレス可能なレジスタに付加

的なハードウェア・レジスタを割り当て、上記順序から外れた命令の前後のアドレス可能なレジスタの値を保持するようにする論理装置とを有する計算機システム。

3. 発明の詳細な説明

A. 産業上の利用分野

本発明は計算機の中央演算処理装置中のアドレス可能なレジスタの管理に関する。より具体的には、本発明は、順序から外れた命令を処理し且つ分岐命令及び割込み時にレジスタ内容の回復処理を行なうためのレジスタ・アレイ用の制御システム及びモニタ・システムに関する。そのようなシステムは、命令のシーケンスが終了した時にアドレス可能なレジスタの新旧の内容を両方共保持する必要がある。アドレス可能なレジスタは、汎用レジスタ及び浮動小数点レジスタを含み得るがそれには限定されない。本発明の実施例は、固定した数のアドレス可能なレジスタの機能に役立つ複数の物理的アレイ・レジスタを有する、IBMシステム

/370のアーキテクチャに適合した計算機プロセッサに関して説明する。

B. 従来技術

典型的な計算機システムの設計は、プログラマがその機械のためにプログラムを設計する時に使う汎用レジスタ(GPR)等の固定数のアドレス可能レジスタを設ける事を必要とする。一覧システムが利用可能になると、アーキテクチャ的に利用可能なGPRの数を変更する事は、新しい数のGPRを利用するため大幅なプログラムの書き直しを必要とする。

同様に、計算機及び計算機プログラムの設計は、計算機プログラムの命令が、それらが書かれシステムに入れられた順序で計算機により実行されるという仮定に基いている。命令は、論理的には、順番に実行されたように、計算機システムに見えなければならないが、依存関係が他の命令との間に存在しないと仮定すると、ある命令は物理的に順に実行される必要はない事が、計算機の性能を

オーバーラップ動作を実現できる制御論理を実現する必要がある。オーバーラップを実施できる多くの異なる形式が存在し、各々はそれ自身の独創的制御プログラムの組を有している。

普通の形式のオーバーラップ技術は、いわゆるバイブライン方式である。非常に単純化すると、バイブルайн方式の機械は、命令の処理の異なる段階毎に別個のハードウェアを設ける。命令が1つの段階の処理を終えると、それは次の段階に移り、次の命令が、丁度空になつたその段階に来る。そのような機械では、たとえ異なる命令に関して異なる処理段階が同時に起きるとしても、その処理のある特定の段階に関しては命令は順番に保持される。そのようなプロセッサにおいては、未だ生成されていない結果が他の命令で必要な事を制御部が検出すると、制御部はその結果が得られそれが必要なところに渡されるまで、バイブルайнの一部を止めなければならない。この制御論理は時々複雑になりうるが、命令がバイブルайн中に順番に保持されるという事実は、その複雑性

改善するための努力において判明した。さらに、ある命令が順序通りに実行されず、且つそのような命令が分岐命令であり、命令シーケンスを選択するために分岐予測が行なわれている場合、間違った分岐が予測されたならば関連のレジスタを元の値に回復する必要が生じ得る。そのような場合、システムは分岐が起きた地点まで回復される。順序通りでない命令を効率的に実行するには、命令によって影響を受けるGPRに関して確定した古い値が保持されると共に、影響を受けたGPRに関して新しい値を臨時に受け取る手段が設けられる必要がある。介在した命令が終了し分岐命令が解決される時、臨時性は除去され、新しい値が、確定値になる。

大型プロセッサは、複数の命令が同時に種々の実行状態にあるようなオーバーラップ技術を、多年にわたつて、用いてきた。そのような技術を用いると、命令間の依存関係を検出し且つ得られた結果が「1度に1命令」のアーキテクチャ・モデルによつて記述されるものになるよう通常のオ

を割り切れる範囲内に保つ上で、確かな助けになる。

プロセッサが別個の実行ユニットを含む場合には、より複雑な形のオーバーラップが生じる。より一般的でないが、この技術も公知であり、多年にわたつて使われている。異なる命令は異なる実行時間を有し、かつ命令間の依存性は可変なので、そのようなプロセッサにおいては、命令が実行され、プログラム中の順序とは異なる結果が生じる事は不可避である。そのようなプロセッサが論理的に正しく動作するには、バイブルайн構成よりも複雑な制御機構が必要である。

しかしながら、従来技術の多重実行ユニットは、任意の時点で正確な割り込みが起きるのを許さない。例えば、命令がオーバーフロー状態を生じるとき、これが検出される時までに、プログラム中の後の命令が既に実行されその結果がレジスタ又は主記憶に置かれる事がある。これにより、割り込みをかけ、以前の全ての命令が実行されその後の命令が実行されていないプロセッサの状態を保

存する事が不可能になる。この例では、オーバーフロー割り込みは、それが起きた時よりも後で実際に認識されるであろう。他の同様の状況も従来技術において存在しうる。

ある従来技術の機械の設計者は、ある実行状態の全ての命令が可能な限りその実行を終了し、次に最近に命令がオーバーフロー状態を有している事を報告する「不正確な」割り込みを起こす事を可能にする事によつて、この状況を処理する事を選択している。これはオーバーフローのような状態に関して割り込みを取り扱う妥当な方法である。この場合、結果はプログラムに返され、プログラムはバグを直すか又は入力データを訂正して、プログラムを最初から再度走行させる。しかし、これはページ・フォールトのような割り込みを取り扱うには許容できない方式である。この場合、システム・プログラムは何らかの訂正動作を行ない、そして割り込みの地点から実行を再開する。

米国特許第4574349号では、各GPRに付属して付加的なレジスタが設けられ、ポインタ

値の使用によりレジスタの名前の変更が起きる。しかし、この特許は、シーケンス外の実行中の間違つて推進された分岐又は割り込みからの正確な回復の問題を解決していない。

IBM Technical Disclosure

Bulletin、1981年8月号、1404~1405頁の論文は、サブルーチンの切り換え時に記憶装置を使用する事を避けるために複数のGPRセットの間で切り換えを行なうシステムを示している。またIBM Technical Disclosure Bulletin、1982年6月号、86~87頁の論文は、命令実行中にディマー・レジスタを使う事を示している。実行が終了すると、レジスタは、結果を受取るために命令によつて名付けられたレジスタとして名前を変更される。実行中、レジスタはトランスペアントであり、これは付加的な物理的レジスタを許容する。しかし、これらの文献のいずれもシーケンス外の命令の実行を取り扱っていない。

IBM Technical Disclosure

Bulletin、1986年8月号、991~99

3頁の論文は、条件付き分岐の解決の間に元のGPRの内容を保存し必要ならばシステム状態を回復するための、1対1に対応した第2のGPRのセットを示している。レジスタの状態を調整し又はレジスタの元の内容を回復するためICPRと共に条件モード・タグが使われている。

C. 発明が解決しようとする課題

本発明は、計算機のプロセッサに付属するアドレス可能レジスタのためのレジスタ管理システムを提供する。このレジスタ管理システムは、命令のシーケンス外の実行のために設けられ、命令がシーケンスからはずれる場合の割り込み又は間違つた分岐予測から正確に回復するための機構を含んでいる。

D. 課題を解決するための手段

本発明は、固定数のアドレス可能レジスタを有するアーキテクチャのプロセッサを想定している。典型的なシステムは、例えば、IBMシステム/370アーキテクチャと同形であり、下記の実施例は主としてそのアーキテクチャにおけるGPRを取り扱う。

本発明は、アーキテクチャによるレジスタの数よりも大きな数のレジスタを有するレジスタ・アレイ(RA)を提供する。実際には提供されるレジスタの数は可変であり、例えばアーキテクチャによるレジスタ数の2倍である。

計算機プログラムの命令はアーキテクチャ中のアドレス可能レジスタの使用を要求するので、RA中のレジスタは、システム/370のGPRのようなアドレス可能レジスタの機能を実行するよう割り当てられる。また命令は命令識別子(ID)数も受け取る。I IDの循回的回転を使用してもよい。またRA中の各レジスタ毎に記入項目を有するアレイ制御リスト(ACL)が設けら

れる。A C L中の各位置は、関連するレジスタに関するいくつかのステータス・フィールド、例えばレジスタの利用可能性状態を含むフィールド、レジスタに与えられたI I D及びレジスタに与えられたG P Rの名前等を有している。システム・アーキテクチャのために、R A中のレジスタがG P Rとして一度割り当てられると、それはプログラムにとつて、同じG P R番号を有する恒久的な物理的レジスタと同じように見える。

またレジスタ管理システムは、R Aに関連し各G P R毎に1つの位置を有する、デコード・レジスタ割り当てリスト(D R A L)及び1つ以上のバックアップ・レジスタ割り当てリスト(B R A L)を含んでいる。D R A L中の各位置は、関連するG P Rに割り当てられたR Aの位置の番号を含んでいる。各命令がデコードされる時、それが参照するG P RがD R A L中で探索され、命令により参照されたG P Rに対してどのR A位置が割り当てられたかが決定される。G P Rのための結果を受け取るために新しいR A位置が割り当てら

停止する。

A C L及びD R A Lは、命令が解説され実行される時に協働して働き、プログラムにより認識される資源であるアーキテクチャ上のG P Rに従つてR Aの内容を管理する。新しい命令が解説される時、それが参照するレジスタは、どのR A位置がそれに割り当てられたかを見つけるためにD R A L中で検索される。その後、R A位置アドレスがG P R名の代りに実行ユニットによつて使用される。R Aの割り当てがD R A Lにおいて判明した後、ステータスを決定するためにA C Lがアクセスされ、その情報が実行ユニットに送られる。

命令が終了すると、そのI I Dは、R A中のI I Dと比較するために実行ユニットによりA C Lに送られる。同じI I Dから結果を受け取つた各R A位置毎に、終了ステータスを表現するように制御タグが変更される。

条件付き分岐に出会うと、命令は、予測された分岐の方向で解説される。直々に命令を完了するという要求があるので、取つた分岐が解決される

れると、D R A Lは新しい割り当てに関して継続的に更新される。

プログラム実行中の正確な地点におけるD R A Lのステータスを凍結し保存し、必要な時にその正確な地点までD R A Lを回復するためには、D R A Lに対して1つ以上のB R A Lが存在する。条件付分岐に出会いと、その地点のD R A LがB R A Lにコピーされる。もし第2の分岐に出会つた場合、B R A Lが存在していれば、その地点のD R A Lが2番目のB R A Lにコピーされ、又それが存在しなければ、それは抜けられる。3番目のB R A L、又はそれ以上の数のB R A Lを、所望により設けててもよい。各B R A Lは、システムが処理を既行している間、特定の固定した時点のシステム・ステータスを保存するのに役立つ。設けられるB R A Lの実際の数は、同時に進行し得る、初期のシステム・ステータスの回復を必要とする可能な状況の最大数に対するシステム設計者の認識に基いて与えられる。もし十分なB R A Lが与えられていないと、条件が解決されるまで実行が

前にプロセッサは、予測分岐中のどの命令に関しても完了信号を出さない。必要であれば、分岐予測が行なわれた後で新しく割り当てられた各R A位置に関して、そのような割り当てが無効化されるよう各A C L位置中の制御フィールドがセットされる。

分岐予測が間違つていた時の分岐回復技術は、プロセッサの全ての部分が関係する。レジスタ管理プロセスに関してそれが意味するものは、分岐点に到達した後で命令の解説が停止した場合にG P Rがそうであつた状態にG P Rのステータスを回復する必要がある事である。このプロセスは、分岐が解説されて以来、G P R制御ステータスに対して2つの型の更新が行なわれている事を認識する。1つの型は分岐以前の命令の、完了に向つての進行及び実際の完了を反映し、その更新の効果は保持されなければならない。第2の型の更新は、分岐後の命令の解説及び実行を反映し、これの更新はG P Rのステータスから除去しなければならない。

(割り込みを除いて) D R A Lは命令が解説される時にだけ更新され、完了によつては影響されないので、D R A Lの内容は、分岐後に命令が全く解説されなかつたならば、変化していないであろう。従つて、D R A Lに関して望ましい事は、分岐命令の解説直後のステータスに、それを回復する事である。これを行なうために、もし2つ以上のB R A Lがあれば適当なB R A LをD R A Lに回復する事によつてR Aの割り当てを正しいステータスに戻す。条件付き分岐が解説される毎に、分岐解説直後のD R A Lの内容はB R A Lに移される。分岐予測が解決される時、B R A Lは廃棄されるか又はD R A Lを回復するために使われる。

割り込み制御は、割り込み地点より先の命令が完了する事を防止するために設けられる。割り込みは、その原因になつた命令の完了又は抑止のいずれを要求する事もできる。先行する命令は、特定の割り込みによつて許される地点までに完了する事が許される。この地点で、D R A Lは、割り込みなしに進行したエントリを含んでいるので、

E. 実施例

本発明は、汎用レジスタのような特定の複数のアドレス可能レジスタ、例えばn個の汎用レジスタ(G P R)というアーキテクチャ上の設計要求を有する計算機システムのためのレジスタ管理システムに関する。n個のレジスタ(但し mはnよりも大きい)を有するレジスタ・アレイ(R A)が、n個の汎用レジスタの機能を実現するために設けられる。説明のための実施例として、16個のG P Rを有する周知のIBMシステム/370アーキテクチャに従うシステムについて説明する。本発明によるR Aは、アーキテクチャ上のレジスタの機能を実現するために、R A位置の動的割り当てを行なう。具体的レジスタ割り当ての機能が終了すると、R A中の位置が解放され、同じ又は他のアーキテクチャのG P Rとして再割り当て可能になる。

本発明のレジスタ管理システムは、全体的な計算機アーキテクチャに依存せず、種々の環境で実

現状に合わない、又は不適切な状態にある。しかし、A C Lは、割り当てられた状態の全てのR A位置に関する正しい情報を含んでいる。A C Lの位置は、キャンセルされ、割り込みを越える命令に関して利用可能なステータスに戻される。次にA C Lは、割り込みから回復するためにD R A Lに現在のステータス値を与るために使われる。

要約すると、本発明のレジスタ管理システムは、順序外の命令及び分岐命令を、R A及び2重機能制御システムを用いて取り扱う。制御システムの第1の部分、D R A Lはアーキテクチャ上のG P Rの視点から命令を管理する。制御システムの第2の部分、A C Lはレジスタ・アレイの実際の内容を管理する。これにより、命令がシーケンス外に実行されても、分岐条件又は割り込みが回復できる。

施できる。例えば、第1A図及び第1B図に示す計算機システム10は、キャッシュ・メモリ・システム14の接続された主記憶12を有する。キャッシュ・メモリ・システム14は、多くの許容可能な方式で構成できるが、この例では、各々命令及びデータを別々に扱う命令キャッシュ16及びデータ・キャッシュ18から構成されている。カスクード構成においてメモリ・スピードとメモリ・サイズの利点を両方共提供するために2レベル以上のキャッシュ・メモリを設ける事はメモリ設計の分野で公知であり、ここに示されてはいないが、そのようなメモリ設計も本発明に適合する。

命令は、命令キャッシュ16から命令バッファ・ユニット20を経由して命令レジスタ・ユニット22へ伝達される。説明のため、命令レジスタ・ユニット22は2個以上の独立した命令レジスタを有し、2、3又は4がそのような命令レジスタの望ましい数である。

計算機設計の分野において、システムが2以上の汎用実行ユニットを持つ事は公知である。例え

ば、汎用ユニットは、実行される機能の型の系列に沿つて、算術又は論理演算、スカラー又はベクトル、スカラー又は浮動小数点、等と設計し得る。汎用実行ユニットのどのような構成も汎用レジスタを利用するので、本発明は計算機中の汎用実行ユニットの数、機能構成及び設計の多くの変型に適用可能である。

説明のため、このシステムは汎用実行ユニット(GPE)1及び2(各々参照番号24、26)を有するものとする。汎用実行ユニット24は出力が記憶バッファ・ユニット28に接続され、これはさらにデータ・キャッシュ18に出力が接続される。汎用実行ユニット24は、実際には、単一の実行ユニットでも又ユニットの組み合せでもよく、この実施例に示されているように、ユニット24は記憶バッファ28に行く結果を生成する。その結果は、命令が完了するまでそこで保持され、次にメモリに記憶される。汎用実行ユニット26は、出力が、汎用レジスタ・アレイ(RA)30に接続される。GPE26は、即座に記憶される

のではなくレジスタで利用可能になる必要のある結果を生成する命令により動作する。命令レジスタ・ユニット22から命令を受け取りそれらをGPE24又は26に適当に振りむけるために命令スタック又はキュー31が設けられる。複数の種々の型の実行ユニットを、单一のレジスタ・アレイ及びレジスタ管理システムと共に用いてもよい。

RA30は、この実施例のアーキテクチャにより認識される16個のGPRの機能を実施するために32個の動的に割り当てられる実レジスタを有する。

RA30は制御バス34を経由してレジスタ管理システム(RMS)32により制御され、且つそれにステータス情報を与える。RMS32は、種々の型のステータス情報を受け取り且つ与えるために種々の他のシステムに接続されている。割り込み制御部36は、割り込みを適切に処理し必要なステータス情報を保存するために、命令レジスタ22、RMS32、及びRA30に接続されている。

RMS32は、命令の発行から実行まで追跡するため並びに入力オペランド及び出力オペランドのためのレジスタ割り当てのために、命令レジスタ・ユニット22並びにGPE24及び26に接続される。

この実施例の計算機は、命令レジスタ・ユニット22から命令を受け取るように接続され、命令アドレス計算部(I-ACE)52に出力を与えるキュー50を有する。I-ACE52はRA30から直接入力を受け取るようにも接続され、また命令キャッシュ16に接続された出力を有する。命令キュー50はステータス情報を与えるためにRMS32に接続される。

この実施例の計算機は命令レジスタ・ユニット22からの出力を受け取るよう接続されたアドレス・キュー60を有する。アドレス・キュー60の出力は、データ・アドレス計算部(D-ACE)62へ入力として接続される。D-ACE62への他の入力はRA30からのものである。D-ACE62はステータス情報を与えるためにRMS

32に接続される。

D-ACE62の出力はアドレス取り出しキュー64に接続され、これはさらに第1の出力がデータ・キャッシュ18への入力として、及び第2の出力がアドレス記憶キュー66への入力として接続されている。アドレス記憶キューは、出力がデータ・キャッシュ18に接続され、且つステータス情報を与えるためにRMS32に接続を有している。

この実施例の計算機は、浮動小数点演算ユニット70を有し、これもステータス情報を与えるためにRMS32に接続されている。RMS32は、RA30に関係付けられないレジスタ及びユニットと共に動作できる事に注意されたい。例えば、1つのRMSは2以上のレジスタ・アレイと共に動作し得る。より具体的には、1つのRMSは、同じ又は異なるたつの型の複数の実行ユニットに接続された2個のRAを制御し得る。

浮動小数点ユニット(FPU)70への入力は、浮動小数点命令キュー72及び浮動小数点データ

・レジスタ・ユニット74によって与えられる。浮動小数点命令キュー72はI-REG22から入力を受け取る。浮動小数点データ・レジスタ・ユニット74はFPU70及びデータ・キヤツシユ18から入力を受け取る。浮動小数点ユニット70の出力は、記憶バッファ・ユニットに接続され、これはさらにデータ・キヤツシユ18に接続されている。

第2図を参照すると、レジスタ管理システム2の詳細な構造が示されている。デコード・レジスタ割り当てリスト(DRAL)100が、ステータス及び制御信号盤に接続される。また論理ユニット101がDRALの内容をモニタし制御するためにはステータス及び制御信号盤に接続されている。DRALは命令が解読され、GPR代入がRA代入に翻訳される時に使われる。DRALは、いくつかの異なる方式で構成できる。例えば、各DRALに複数コピーを有する2個以上のDRALが存在し、それが各GPR毎に一つの位置を含み、その位置が、そのGPRに関して値を受け

ALはDRALと同じ構造を有し、1サイクルでDRALの全内容がBRALにコピーできまた逆も可能なように接続されている。これらの転送は、論理ユニット101によって制御されている。それは、例えば、分岐が起きる方向に付する予測が間違っている場合にDRALの内容を保存するために、条件付分岐に出会う時に使用される。

各DRAL毎に1つだけのBRALが設けられている場合、通常、1つだけの条件付分岐を過ぎて解説を行なう事しか可能でない。しかし、GPRを変更する命令が間に介在する事なく第2の条件付き分岐命令に出会うという特別な場合には、その分岐命令も過ぎて解説する事が可能になる。というのは同じDRAL内容が両方の分岐に関してBRAL中に保存されるからである。

アレイ制御リスト(ACL)110は、ステータス情報を受け取り且つ制御情報を送るためにRA及び計算機システムの残りに接続される。論理ユニット101はACL110の内容を制御し、ACL及びDRALの動作を調整する。GPRを

取るように最も最近割当てられたRA位置の数を含んでいる。各命令が解説される時、それが参照するGPRがDRAL中で検索され、どのRA位置がそのGPRに割当てられたかが決定される。また、新しいRA位置が結果を受け取るように割当てられる時、それらの割当てを反映するようにDRALが更新される。このようにして、GPRを使用する各命令が、そのGPRを最も最近に参照した命令に割当てられたRA位置を見出すようICRALによつて指示される。

バックアップ・レジスタ割り当てリスト102、104、及び106は、動作のある特定の時点でのDRAL100の内容全体を受け取るように接続される。通常、各DRALに対応してシステム中に少なくとも一つのDRALが存在する。もしシステムが分岐の解決を待機する事が許されるならば、レジスタ管理システムはBRALなしに動作しうる。一つ、二つ又は、三つのBRALを使用すると、それぞれ待機なしに一つ、二つ又は、三つの条件付分岐を処理する事が可能になる。BR

サポートする各RA毎に、そのRAに関するステータス情報を記憶するACLレジスタが存在する。アレイの各レジスタ位置毎に1つのエントリが存在し、この実施例では、各エントリは第3図及び第4図に示すように、CTL、ABC、IID、PRV及びREGの5つのフィールドに分割された14ビットから構成される。CTLはRA位置の全体的ステータスを定義する制御フィールドである。それは下記の値を取り得る。

00：利用可能—そのRA位置は使用中ではなく、必要であれば割り当てる事が可能である。CTL=“00”的時、ABC、IID、PRV及びREGのフィールドは何の意味も持たない。

01：割り当て済—このRA位置はREGフィールドにより指定されたGPRに割り当てられている。完了した現在の命令に割り込みが起きると、これはそのGPRに対応するRA位置である。任意の与えられた瞬間には、各GPRに割り当て1つだ

けのRA位置が割り当てられている。CTL = '01' の時、ABC フィールドは '000' であり、IID 及び PRV フィールドは意味を持たない。

10：係員中且つ未ロード—このRA位置は、まだ実行が完了していない命令の結果を受け取るように割り当てられているが、これまで命令は結果をこのRA位置にロードしていない。IID フィールドは、このRA位置が結果を受け取る命令に割り当てられた IID である。ABC フィールドは、この命令が条件付きで実行されれば非ゼロであり、さもなければ '000' である。多くの場合、REG フィールドは、このRA位置が結果を受け取る GPR の番号であり、PRV フィールドは、その GPR の古い値を保持する RA 位置の番号である。GPR を変化させない比較命令という特殊な場合では、RA 位置にはとにかく割り当てが行なわ

設けた。

3つまでの条件付き分岐に出会つてもよいように3つのBRALが設けられているこの実施例の場合、ABC フィールドは3ビットを有し、これは最初 '000' にセットされる。最初の分岐に出会うと、第1のBRALに DRAL の内容がロードされ、ABC フィールドは、BRAL のロード及び未解決の分岐の存在を示すために '100' に変更される。第1の分岐が解決される以前に第2の分岐に出会うと、第2のBRALにその時点のDRAL の内容がロードされ、ABC フィールドは '110' に変更される。最後に、最初の2つの分岐が未解決で且つ第3の分岐に出会うと、第3のBRAL が、その時点のDRAL 内容を受け取り、ABC フィールドは '111' になる。ABC フィールド中の各ビットは、BRAL が活性であり、未解決の分岐に関する特定の DRAL 内容を記憶している事を示すために独立に特定の BRAL に関係付けられている。論理ユニット 1 01 はこれらの機能を制御する。

れ、REG フィールドは無関係であり、PRV フィールドはとのRA位置の番号を含む。

11：係員中且つロード済—このRA位置は、まだ実行が完了していない命令の結果を受け取るように割り当てられているが、その結果を受け取っている。ABC、ID、PRV 及び REG のフィールドは、CTL = '10' の時と同じ意味を有する。但しこの状態では比較命令に関する特殊な場合は起きない。

ABC フィールドは、条件付き分岐を過ぎて解説された命令に関する RA 位置を識別するために使われる。この情報は IID を調べる事により決定できるので、本発明を実施するためにそのようなフィールドは必ずしも必要ではない。しかし、IID をテストする事により定期的にそれを決定するよりも、この少量の情報を記憶するために専用のフィールドを設ける方が、より経済的且つ高速であるので、この実施例ではこのフィールドを

もし分岐が好ましい方向で解決されると、割り当てられた BRAL 内容は不要であり、ABC フィールドの適当なビットが 0 にセットされる。以前の例を参照すると、分岐は必ずしも順序に解決される必要はない。もし第1の分岐が起き (ABC = '100') そして次に第2の分岐が起き (ABC = '110') たとすると、第2の分岐が最初に解決され、ABC フィールドが単に '100' にリセットされてもよい。もし最初の例のように、3つの分岐が順序に起き (ABC = '111') 、そして第2の分岐が最初に解決されると、第2のBRAL が利用可能になり、ABC フィールドは '101' にセットされる。さらに、これは、他の分岐に出会った場合に1つのBRAL が利用可能である事を意味する。ABC フィールド中のビットは、分岐の発生及び解決に対応して任意の順序でセット及びリセットできる。例えば、ABC フィールドが '101' であれば、新しい分岐に出会つて 0 ビットがセットされ、ABC フィールドが (これは分岐の発生した順序を

表していないが)「111」になる事もある。

命令が解読される時、それが参照する G P R は D R A L 中で検索され、どの R A 位置がそれに割当てられているかが見出される。これは、使用されている G P R 及び変更された G P R の両者を含む。もし命令が G P R の値を変化させると、新しい値を受け取るよう R A 位置が割当てられ、これらの新しい R A 位置がこれらの G P R に関連付けられるように D R A L が更新される。次に、R A 位置の割当では、G P R の実際の割当ではなく実行論理ユニットに伝達される。

プロセッサが二つの命令を同時に解読する能力を有していると仮定すると、D R A L は、それらの各々に関して、R 1: R 1 + 1、X 及び B レジスター (IBM システム / 370 の命令形式を参照されたい) を検索する能力を提供する。一般にアーキテクチャを参照すると、命令は、必要なメモリ・アクセスを行なうために、インデックス値及びベース値を求めて G P R を要求する事がある。これは多くの命令に関して適切であるが、適切で

と、それは「保険中且つ未ロード」状態 (C T L = '10') にセットされ、I フィールドは割り当てが行なわれた命令の I D にセットされ、R E G フィールドは G P R の番号にセットされ、P R V フィールドはそのレジスタに以前に割り当てていた R A 位置の番号 (D R A L 中を検索する事により決定される) にセットされる。但し比較命令の場合は、P R V フィールドは、ちょうど割り当てられたこの R A 位置を指すようにセットされる。比較命令に R A 位置を割り当てる理由及びこの動作方式の理由は、後述する。

R A に関するこの制御構造は、命令の解読にある条件を隠す。

1. D R A L 中でレジスタを検索する機能は大多数の命令の要求を満足するが、異なる要求を有するものが存在する。これらの命令は解読に 2 サイクル以上を要する。システム / 370 のアーキテクチャにおけるそのような命令の部分的なリストは、M V C L、C L C L、A X R、S X R、L M、S T M、E D M K 及び T R T である。

ない場合、解読に複数サイクルが必要である。システム / 370 のアーキテクチャにおいて、複数ロード命令は 16 個までの G P R を参照できる。各 G P R が D R A L 中で検索された後、そのようにして見つけられた R A 位置が、その R A 位置がロードされたか否かを判定するために A C L 中で検索され、この情報が命令と共に実行ユニットに送られる。R A 位置は、「割り当て済」又は「保険中且つロード済」の状態 (C T L = 'X1') であれば、ロード済と考えられる。

プロセッサは、各サイクル毎に少なくとも 2 つの新しい R A 位置を割り当てる能力を提供する。例えば、G P R を、G P R の各群毎に別個の R A を有する 1 つは偶及び 1 つは奇の 2 つの群から構成してもよい。この時、各サイクルに、2 つの R A に対応する G P R の各群から 2 つ、4 つの G P R を割り当てる事ができる。これを行なう回路は、A C L 中の C T L フィールドを調査し、「利用可能」状態 (C T L = '00') にある最初の R A 位置を選択する。もし R A 位置が割り当たる

2. 1 サイクル当たり 2 つ以上の R A 位置をレジスターの各組に割り当てる能力はある状況では重要である。複数の偶 / 奇汎用レジスタ対又は 2 以上の浮動小数点レジスタを変更できる、370 アーキテクチャの命令の部分的なリストは、M V C L、C L C L、A X R、S X R 及び L M である。これらの命令は、1 サイクルに充分な R A 位置が割り当たられなければ、1 サイクルで解読できない。個々の命令の問題を越えて、これは、どの命令が同時に解読可能かという事に對して条件を与える。同じ組の中で余りに多くのレジスターを変更する 2 つの命令は同時に解読する事ができない。

3. D R A L が正しく動作するためには、解読時にレジスター位置が検索される時、そのレジスターを変更した最も最近の先行命令の結果を受け取るように割り当たられた R A 位置に関する番号を D R A L が含んでいる事が必要である。これは、その命令が直前の先行命令であつてプロセッサが同時に両方の命令を解読しようとしているのでなければ、何の問題もない。この状況を取り扱うために、あ

石命令がレジスタを変更し且つ後続の命令がそのレジスタを参照する時は、必ず、2番目の命令は最初の命令と同時に解説される事を許されない。

各命令が完了すると、そのI I DがA C Lに送られる。この命令からの結果をどのR A位置が受け取つたかを判定するために、論理ユニット101によつてA C L中の全てのI I Dフィールドに對して比較が行なわれる。次に、このようにして同定されたR A位置の各々に関して、その状態を「係属中且つロード済」から「割り当て済」に変更する(C T Lを“11”から“10”へ変更する)信号が作られる。また、これらのR A位置の各々に関して、どのR A位置が利用可能とされる必製があるかを決定するためにP R Vフィールドが調査される。P R Vフィールドによつて指示されるR A位置毎に、その状態を「割り当て済」から「利用可能」へ(C T Lを“01”から“00”へ)変更する信号が発生する。

これらの機能を実行する全ての論理は並列に動作する。例えば、1サイクルで、完了した命令に

値を受け取るよう選択された全ての新しいR A位置はA B Cフィールドが、その分岐に割り当てられたB R A Lに關するエントリにおいて“1”にセットされる。後に、分岐の方向が確定した時に、その分岐後の全ての命令の処理を取り消して他の方向で命令の処理を開始する必要があるかもしれない。

実際には、どの型の分岐も全て条件付きとして扱われる。というのは分岐命令のターゲット・アドレスを識別するために分岐履歴テーブル(B H T)が共通に使用されるからである。B H T中のターゲット・アドレスの位置は、少なくとも正しいターゲット・アドレスが決定されるまで、命令を条件付ける。B H Tが最初にターゲット・アドレスを識別する時、ターゲット・アドレスが正しい事は後まで明らかでない。ターゲット・アドレスの正しさが解決される時、分岐の条件付き状態が除かれる。

失敗した分岐からの分岐の回復はプロセッサの全ての部分が関与する。レジスタ管理システムに

関する古いレジスタ値を含む全てのR A位置が利用可能状態に変更され、新しい値を含む全てのR A位置が、割り当て済み状態に変更される。命令の正規の完了は、D R A Lに對して何の影響も持たない。

理論的には、命令自身を調べる事によつて分岐が起きるか否かを知る事ができない場合、その分岐は条件付き分岐と考えられる。このカテゴリーに属する370の分岐命令は、B C(M=0又はF及びR2=0)、B C T、B C T R(R2=0)、B X L E及びB X Hである。これらの分岐命令の1つが解説されると、I要素は、分岐が行なわれると予測すべきか又は行なわれないと予測すべきかの判定を行なう。次に、それは解説に進み、予測された方向で命令を実行する。一方プロセッサは分岐が実際にその方向に進むか否かを決定する。プロセッサが出来ない唯一の事は、これらの命令を完了する事である。というのは、それらは論理的に分岐に続いており、分岐が完了するまで完了できないからである。この期間中、レジスタ

関してそれが意味するものは、命令解説が分岐の後で停止した場合にG P Rの状態を回復する必要がある事である。分岐命令が解説されてからの中間の時期には、レジスタ制御状態に對して2つの型の更新が行なわれる。1つの型は、分岐以前の命令の実行へ向つての進行及び実際の完了を反映する。この更新の効果は保存されなければならない。第2の型の更新は、分岐後の命令の解説及び実行を反映する。この更新は、無効又は無意味であるとしてG P Rの状態から取り除かなければならない。

通常、D R A Lは命令が解説される時にしか更新されず、その完了によつては影響を受けない。従つて、D R A Lの内容は、分岐後に命令が解説されなければ変更されない。割込みの状況は異なつており、別に説明する。従つて、D R A Lに關して望ましい事は、分岐命令の直後の状態にそれを回復する事である。これはB R A Lを使用する事によつて達成される。条件付分岐命令が解説される毎に、分岐命令解説直後のD R A Lの内容が

B R A L に移され、同時に A B C フィールドの適当なビットが '1' にセットされる。それは、分岐に関する予測が正しかつたか又は間違つていたかが決定されるまでそこに保持され、そして戻乘されるか又は D R A L に回復される。

間違つた条件付き分岐の後では、A C L を正しい状態に回復する事が必要である。条件付き分岐が解説された後に行なわれた全ての A C L エントリは、A B C フィールドが、その特定の分岐に関する B R A L のために使われる特定のビット位置において '1' にセットされる。これらのエントリは全て、条件付分岐が解説された時に「利用可能」状態であつたか又は、その後、何らかの初期の命令の完了によつて「利用可能」状態に置かれた R A 位置にあつて、条件付分岐後に何らかの命令により再利用のために取り出された R A 位置に関するものである。これらの R A 位置のどれも、「係属中」状態の 1 つを越えて進行する事はできない。というのは条件付分岐以後の命令はどれも完了できないからである。条件付分岐以後、全く

無条件分岐命令は、命令処理がどのように進行すべきかについて何の不確定性も生じず、従つてレジスタ管理システムはそれに対して何の注意も払わない。B A L 及び B A L R はこのカテゴリーに属するシステム / 370 の命令であり、且つ G P R を変更する。従つて、それらはレジスタを変更する他の命令と同様に扱われる。B H T を有する機械において、このカテゴリーの命令は存在しないかもしれない。

割り込みは他の状態とは少し違つたふりに取り扱う必要がある。割り込み条件が検出されるとすぐに、割り込み制御部 36 に信号が送られる。このシステムは、命令完了制御部と通信して、割り込み地点以後の命令が完了する事を阻止する。割り込みの地点は、その原因になつた命令の直前又は直後である。これは割り込みの型が命令の抑止を要求するか又は完了を要求するかに依存する。割り込み地点に先行する命令は完了する事を許される。その地点において、D R A L の内容は、割り込みの原因になつた命令以後のいくつかの命令

命令が解説されていなければ、これらの R A 位置の全ては「利用可能」状態にあり、それはそれらが戻されるべきものである。

特定の分岐に関して A B C フィールドが '0' にセットされた R A 位置は、「割り当て済」状態にある R A 位置の組及び、条件付き分岐に先行する係属中の命令に関連するそれらを含んでいる。分岐以後に解説が行なわれなければ、これらのエントリは同じ状態のままであり、従つてそのままにされるべきである。従つて、間違つて予測された条件付分岐の後に A C L を正しい状態に回復する手続きは、分岐に関する特定のビット位置において A B C = '1' であるような全ての A C L エントリを「利用可能」状態 (C T L = '00') にセットする事である。もし条件付分岐が正しく予測された事が判明すれば、A C L 中の全ての A B C ビットはその分岐に関する特定のビット位置が '0' にセットされる。

分岐は、解説時にそれが起きる事が視察により決定できるならば、無条件であると考えられる。

が実行された場合に取られたであろう動作を反映する。

A C L は、「割り当て済み」状態にある全ての R A 位置が、割り当てられた G P R IC に関する正しい値を含んでいるものであるような状態にある。さらに、A C L は、「係属中」状態の 1 つにある R A 位置の数を有していてもよい。これらは全て割り込み地点以後の命令に関連しており、次のステップはこれらの R A 位置の全てが「利用可能」状態に戻される（もし C T L = '1X' ならばそれは '00' IC にセットされる）事である。A C L は正しい状態にセットされ、そして D R A L を対応する状態にする必要がある。

D R A L は、G P R アドレスの各々を経てカウンタを歩進する事によつて数サイクルの期間にわたつてセットされる。各サイクル毎に新しい値を経てそれが歩進する時、その値は A C L エントリの各々と比較される。この機能は論理ユニット 101 によつて達成できる。もし R E G フィールドがカウンタ中の値に一致し且つ R A 位置が「割り

当て済」状態($CTL = '01'$)であれば、比較一致が検出される。各 GPR に対して正確に 1 つの RA 位置が割り当てられなければならないので、この比較プロセスは各サイクル毎に正確に 1 つの RA 位置に対して比較一致を発生しなければならない。これらの比較の結果は RA 位置の番号を発生するように符号化され、これらの RA 位置の番号は DRAL に送られ、順々に DRAL エントリ上に組込まれる。このプロセスの様りに、DRAL エントリの各々は、そのエントリが各々対応する GPR に割り当てられた RA 位置を指示する。これは DRAL 上に記述する正しい状態である。

DRAL をその正しい状態に回復するプロセスは、割り込みプロセスに余分の時間を付加しないようプログラム・ステータス・ワード(P SW) の交換のプロセス中に実行する事ができる。

F. 発明の効果

本発明を用いれば、条件付き分岐や割り込み等の、通常の処理順序から外れた命令の処理を効率的に行なう事ができる。

4. 図明の簡単な説明

第 1 A 図及び第 1 B 図は本発明による計算機システムの実施例の概略図。

第 2 図は第 1 A 図及び第 1 B 図に示した実施例の中のレジスタ管理システムの図。

第 3 図は第 2 図の中のアレイ制御リスト(ACL) の図。

第 4 図は第 3 図に示した ACL のエントリのフィールド構成を示す図である。

出願人 インターナショナル・ビジネス・マシンズ・コーポレーション

代理人 弁理士 順 宮 孝一
(外 1 名)

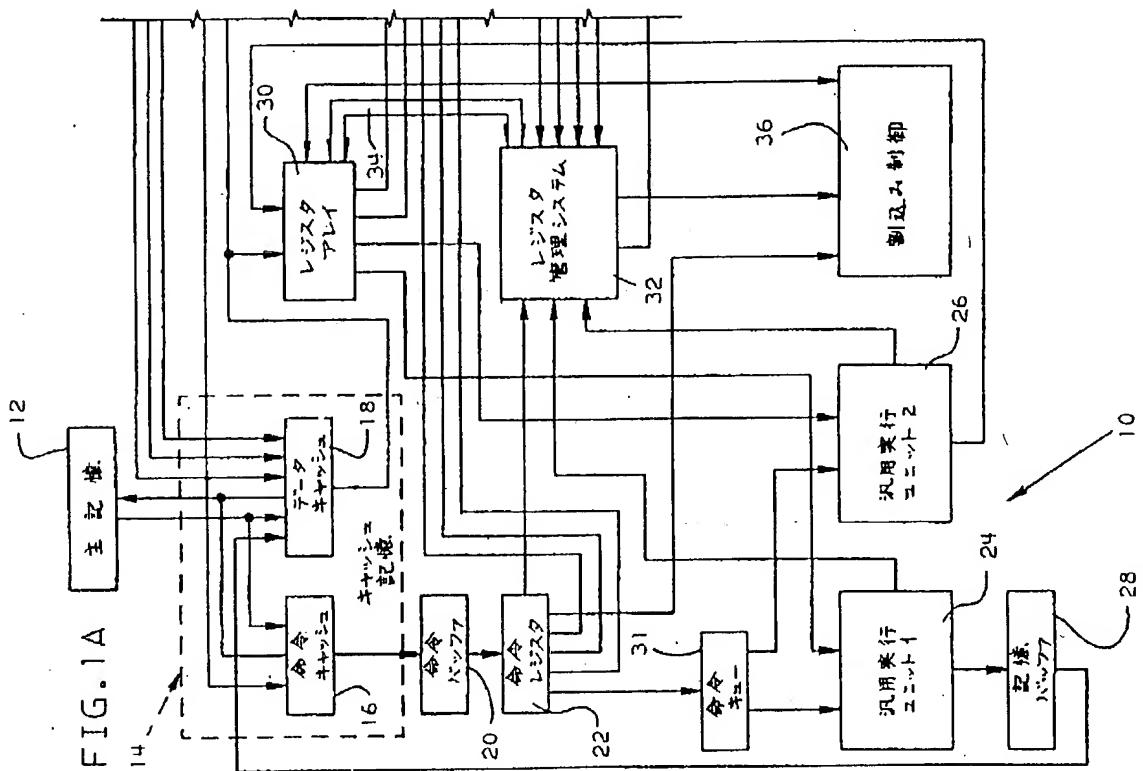


FIG.1B

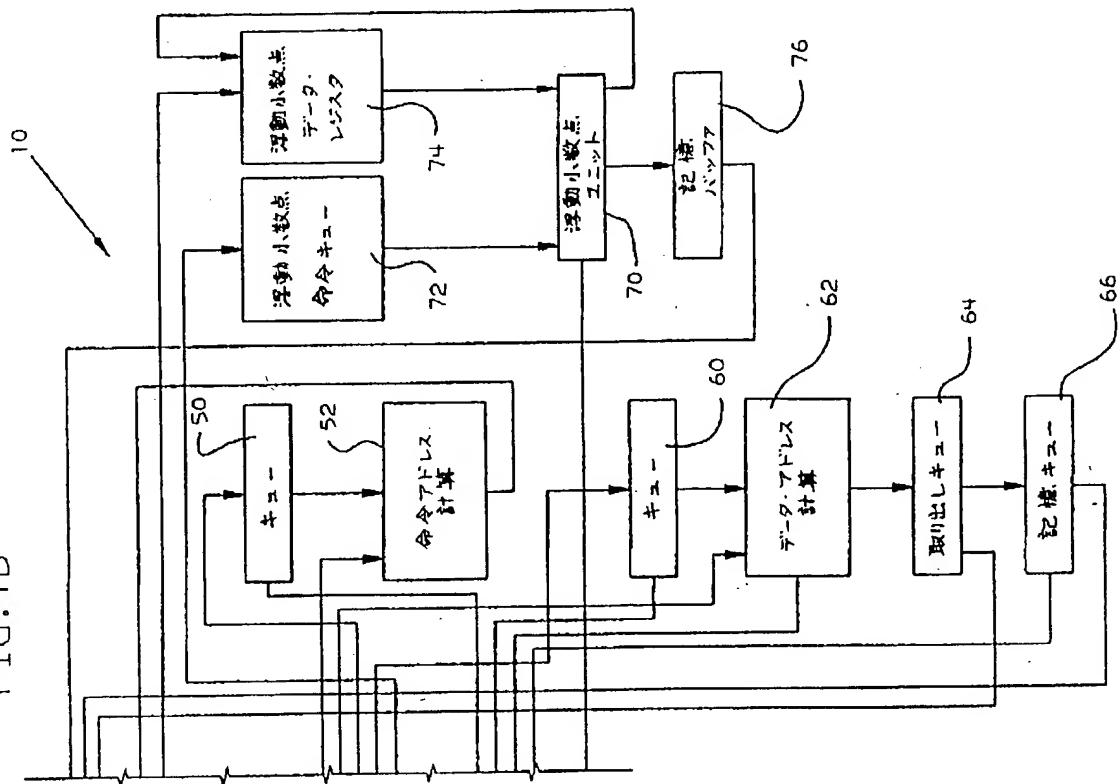


FIG.2

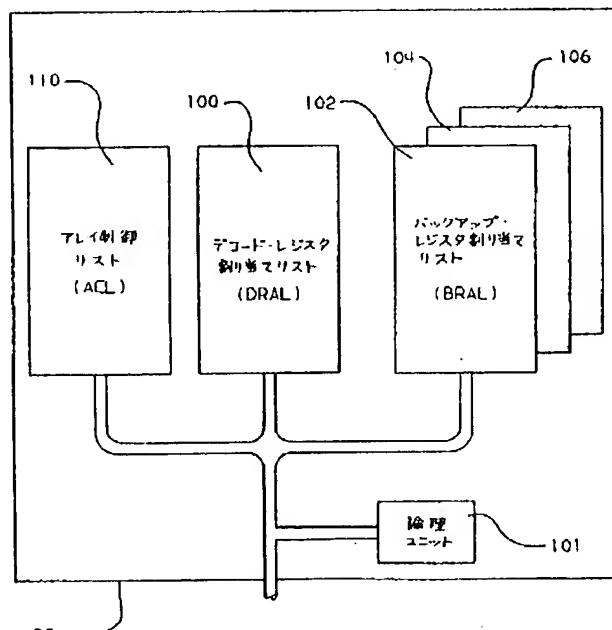


FIG.4

CTL	ABC	IID	PRV	REG
0	2	5	10	14

FIG. 3

0 110

CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG

CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG
CTL	ABC	IID	PRV	REG